SP-1155

Co-occurrence and Dependency Logic for Answering English Questions

3 April 1963

(SP Series)

SP-1155

Co-occurrence and Dependency Logic for Answering English Questions

Robert F. Simmons
Sheldon Klein
Keren McConlogue

April 3, 1963

SYSTEM DEVELOPMENT CORPORATION, SANTA MONICA, CALIFORNIA

CO-OCCURRENCE AND DEPENDENCY LOGIC FOR ANSWERING ENGLISH QUESTIONS*

1.0 Introduction

So far, the digital computer has been truly impressive in its adaptability to a wide range of data-processing functions. Its uses for routine office tasks such as inventory keeping or payroll processing and its application to various military data-processing systems including some verbal information control and retrieval uses has far outstripped direct mathematical and engineering tasks. In the minds of many researchers working on information retrieval, mechanical translation and other language-processing tasks, there is little doubt that computer adaptability will also eventually include the capability of processing natural English text at a level of sophistication now possible only to humans.

One of the most attractive approaches to language processing on computers is the research on program systems that enable the computer to answer natural English questions. The answers are variously proposed to be fragmented lists of information, selections of sentences and paragraphs from text or, most recently (6), the generation of a new statement combining the question with some fragments of text that appear to answer it.

The appeal of this avenue toward language processing lies in the utter generality of the problem of answering questions. Any statement, no matter how complex, can be transformed into a question. Thus any analysis suitable

*This research was conducted under ARPA contract SD-97.

for all possible questions must cover the breadth of the language. In order to find potential answers for questions, techniques for analyzing input data and organizing it in the computer storage must be discovered. So far these techniques lead to interesting analogies with human remembering functions. Finally, the evaluation of potential answering statements or data as actual answers to the question partakes of all the interest and difficulties of studies in the logic of inference and the psychology of problem solving.

Yet for all the breadth that is possible (and eventually necessary) in question-answering systems, they lend themselves readily to fractionation into meaningful small packages. For example, Green (2), by prestoring his language input data into a meaningful structure, was able to concentrate entirely on deriving some of the operations that a question implied for the data store. Earlier, Lindsay (7) had specialized on the problem of where to put a given name in a family tree when his system read a statement about family relationships. More recently, Kirsch* has developed a system for answering yes or no to a limited set of questions about geometric figures in a two-dimensional space.

The Synthex research project of the authors has attempted on a fairly shallow level to treat with many aspects of question answering. Their broad approach is seen as a road toward the development of a general-purpose language processor that will synthesize some of the complex human cognitive processes involved in the task of dealing meaningfully with language. The

*Personal communication; see also (9).

prototype research vehicle, Protosynthex I, accepts ordinary text and English questions as its inputs. It indexes the text and analyzes the questions into the terms of the index. The terms of a question are looked up in the index, and potential answering statements are selected from the text and scored for relevancy to the question. At that point the most challenging task begins--that of evaluating whether or not a statement is an answer to the question.

This paper is designed to show how a question contains many criteria for recognizing its answer and how some of these criteria may be used to find and evaluate potential answering statements. The techniques that are discussed have been developed as working computer programs for various IBM digital computers and are used as part of Protosynthex I.

2.0 The Nature of Questions

One of the most critical aspects of research is formulating a meaningful question. From the scientific point of view, a meaningful question is one for which it is possible to devise a set of operations that can yield answers such as "yes," "no," or "to a certain extent." An example of such a question taken from perceptual work in psychology is: "Can a human, with one eye covered, perceive depth relations between two objects?" The question is complete. It contains all the information necessary to recognize an answer. From the question a set of operations, i.e., an experiment, can be derived which will allow a statement of truth or falsity to be made.

In ordinary English, too, complete questions occur frequently: "Did John go to the store?" "Finished breakfast yet?" "Do worms eat grass?" In every case of such questions the only information lacking is the knowledge of

whether the contained assertion is true or false. Other English questions are less complete, e.g., "Where did John go?" "What do worms eat?" Later in this paper, a close examination of the incomplete question will show that the question words--who, what, where, etc.--contain a great deal of information which is useful in finding and evaluating potential answers.

The first problem in answering a question is to understand what it is asking about. If just the selection of words contained in the question is considered, it can be seen that a vast amount of information is usually present. The words used in a question can be categorized into three large classes: the question words "who," "what," "where," etc.; the function words which indicate grammatical constructions such as articles, prepositions, conjunctions, etc.; and finally the content words such as nouns, verbs, adjectives, etc., which carry the bulk of meaning contained in the question.

In terms of information theory the selection of content words in a question represents a choice of five or ten words from a population of words numbering somewhere between 75 and 150 thousand. The amount of information actually contained in each word is partially a function of its co-occurrence with other words, but is roughly equivalent to the inverse of its probability of selection. Thus the content words selected from a list of the order of a hundred thousand words contain far more information than the function words which are a selection from a list of a few hundred words or the question words which are selected from a list of less than a dozen.

It is the content words on which we must rely most heavily to discover possible answers to the question. They carry the largest share of meaning.

In the system to be described below, the content words are index entries. The index cites the location of all statements in a text that uses these content words and makes it possible to retrieve information-rich statements--not answers necessarily, but data that is pertinent for answering questions. Although they do not contain as much information as the content words, some of the function words and all of the question words are also essential cues for use in evaluating possible answers to questions.

The English question words--the relative pronouns "who," "what," "where," "how," etc.--all carry very special meanings. They are pronouns that substitute for certain semantic classes of words. "Who" substitutes for a person, "where" for a place, "when" for a time and so on, to indicate the type of word or phrase that is required in an answer. In addition to indicating semantic classes required by an answer, they also signify syntactic classes of words. "Who" and "what," for example, show that the answer should be in a nominal or noun-phrase construction. "Where" and "when" in contrast require a verb-modifier construction for the answer. In the latter case, the question words also select a small set of prepositions or adverbs such as "in," "on," "at," "near," etc. "How" is answered by an overlapping set of prepositions though still usually in a verb-modifying construction. These cue constructions include "by means of," "with," "by use of," "by...ing," etc.

By using the information contained in the question words, such an incomplete question as "What do worms eat?" may be transformed into a statement that will help to identify the answer. The first step in this transformation is to change the question back into the structure of an assertion, e.g., "Worms

eat what." (Just how this is actually accomplished by a computer system will be discussed later.) The question can now be further transformed into "Worms eat X; X = thing, nominal." "Where do worms eat" would transform into the following:

"Worms eat X; X = (prep/adv place, and place, nominal)."

If the potential answering statement includes "worms," "eat," and a place word modifying the verb "eat," and all these words are in an appropriate set of relationships then it is known that an answer is present. (Whether or not the answer is true is yet another difficult question still to be dealt with.)

But just exactly what is an appropriate syntactic arrangement of the question words to allow for a possible answer? Such acceptable answers as the following come readily to mind for the question "What do worms eat."

Grass is eaten by worms.
A worm-gnawed apple...
Worms eat their way through the ground.
Worms eat grass and bits of vegetation.

Some unacceptable answers follow:

Birds eat worms.
Horses with worms eat grain.
Worms are eaten by birds.

It is to be noticed that no easy prediction can be made as to just where in an answering statement the things that worms eat are to be found. Truly the answers to a "what" are always in nominal constructions, but that fact is not in itself enough to determine that a statement is an answer. What is pertinent is that the terms in the answering assertion bear the same set of interrelationships as they do in the question.

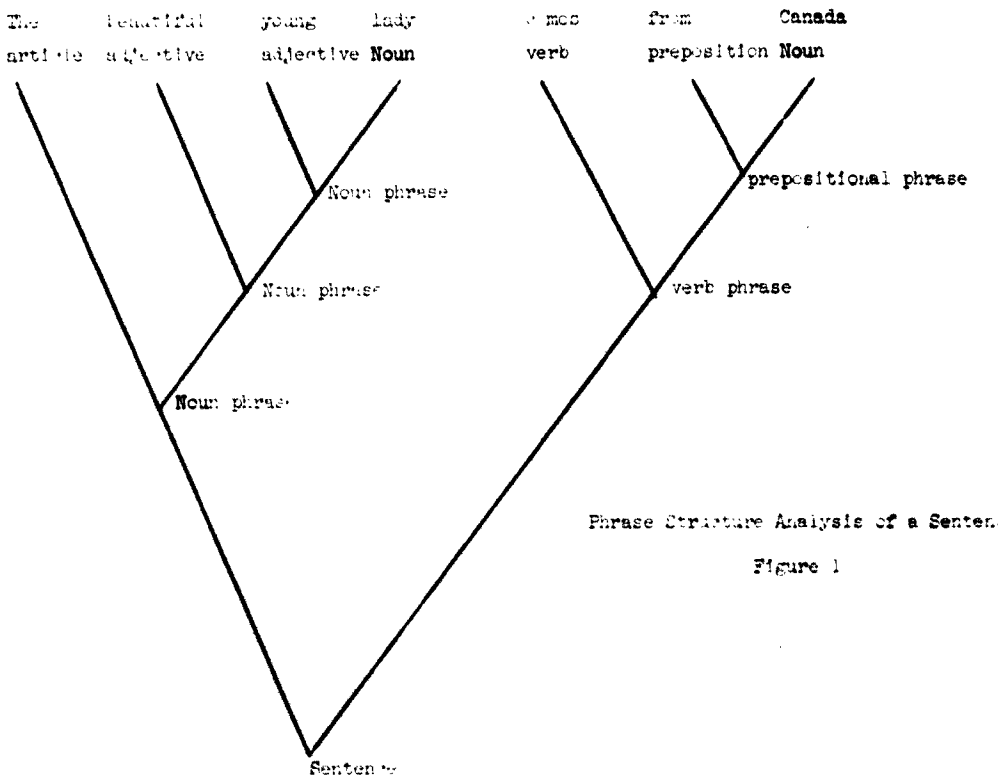
To discover whether the relations are parallel between the question and the potential answer, a detailed syntactic analysis must be made. But by itself the syntactic analysis is not enough. The awkwardness of comparing syntactic structures is well known. Harris (3) and Chomsky (1), for example, have spent many years developing transformation rules which show which syntactic strings are substitutable, one for the other, without seriously changing the meaning of a phrase. There is, however, one aspect of the syntactic analysis which can be compared easily from question to answer. The dependency relations of the answer must be essentially the same as those of the question.

3.0 Phrase Structure and Dependency Analysis

A phrase structure analysis of a sentence depicts relations among units which can consist of individual words or groups of words. A dependency analysis is concerned only with relations among individual words--relations of dependency or modification.

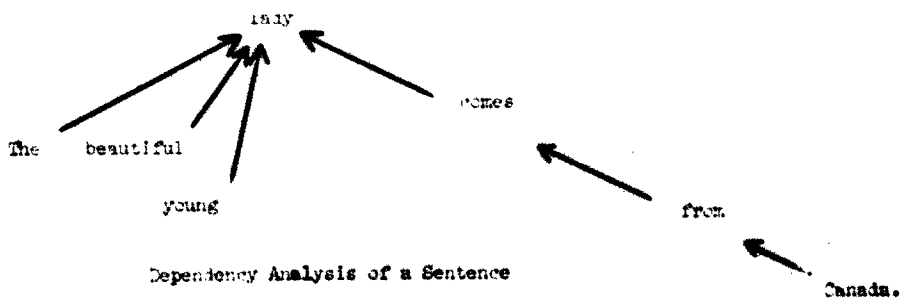
While a dependency analysis can be derived from a phrase structure analysis of a sentence, the reverse is not true. Accordingly, a dependency analysis contains less information, but the information it does contain can be coded in the notation of a single binary relationship, i.e., one word is or is not dependent upon another. Such a feature permits extremely simple computer handling of complex syntactic features.

A phrase structure analysis can be in the form of a tree structure whose nodes are labeled with the names of the types of construction they represent, as illustrated in Figure 1. A dependency analysis may also be in the form of a tree structure, as illustrated in Figure 2.



Phrase Structure Analysis of a Sentence

Figure 1



Dependency Analysis of a Sentence

Figure 2

Note that in Figure 2, "the" and "beautiful" are directly dependent upon "lady" although they are separated from "lady" by intervening items in the actual sentence. The reason for this can be suggested by a brief examination of the notion of a phrase structure generation grammar. (See Chomsky (1) and Yngve (11).) The sentences of Figure 2 can be generated by iterative application of a set of rewrite rules of the following type:

1. sentence = noun phrase + verb phrase
2. noun phrase = article + noun phrase
3. noun phrase = adjective + noun phrase
4. verb phrase = verb phrase + prepositional phrase
5. prepositional phrase = preposition + noun phrase
6. noun phrase = noun
7. verb phrase = verb
8. noun = lady
9. noun = Canada
10. article = the
11. adjective = beautiful
12. adjective = young
13. verb = comes
14. preposition = from

In using these rules, the term on the left of the equal sign is to be rewritten as the term or set of terms on the right. Thus "sentence" is rewritten as "noun phrase + verb phrase." This set of terms is similarly expanded by rewriting each element; thus "noun phrase" is rewritten as "article + noun phrase," etc. Following these substitution rules the example sentence might be produced as follows:

a. sentence
b. noun phrase + verb phrase (by rule 1)
c. article + noun phrase + verb phrase (by rule 2)
d. the + noun phrase + verb phrase (by rule 10)
e. the + adjective + noun phrase + verb phrase (by rule 3)
f. the + beautiful + noun phrase + verb phrase (by rule 11)
g. the + beautiful + adjective + noun phrase + verb phrase (by rule 3)
h. the + beautiful + young + noun phrase + verb phrase (by rule 12)

i. the + beautiful + young + noun + verb phrase (by rule 6)
j. the + beautiful + young + lady + verb phrase (by rule 8)
k. the + beautiful + young + lady + verb phrase + prepositional phrase (by rule 4)
l. the + beautiful + young + lady + verb + prepositional phrase (by rule 7)
m. the + beautiful + young + lady + comes + prepositional phrase (by rule 13)
n. the beautiful + young + lady + comes + preposition + noun phrase (by rule 5)
o. the + beautiful + young + lady + comes + from + noun phrase (by rule 14)
p. the + beautiful + young + lady + comes + from + noun (by rule 6)
q. the + beautiful + young + lady + comes + from + Canada (by rule 9)

Note that at stage d, "the" was immediately adjacent to the noun phrase unit, although it later became separated. Similarly, at stage f, "beautiful" was also adjacent to that noun phrase. It is immediate contiguity of elements at some stage in the generation process that helps to determine the dependency between elements which are physically separated in the final sentence. Note that the phrase structure tree in Figure 1, if turned upside down, also represents the history of derivation as described above.

This discussion of phrase structure and dependency is extremely simplified and ignores criteria for the order of application of the phrase structure rules. A detailed discussion of the relationship between the two, including an algorithm for deriving a dependency analysis from a phrase structure analysis, can be found in the work of Klein (5).

Some rules for deriving a dependency analysis from rather general syntactic criteria are shown below (5, 6). (For a different type of dependency analysis see the work of Hays (4).)

1. The head of the main verb phrase of a sentence or clause is dependent upon the head of its subject.

2. The head of a direct object phrase is dependent upon the head of the governing verb phrase.
3. Objects of prepositions are dependent upon those prepositions.
4. Prepositions are dependent upon the heads of the phrases they modify. Prepositions in the predicate of a sentence are dependent upon the head of a verb phrase and also upon the head of an intervening noun phrase if one is present.
5. Determiners and adjectives are dependent upon the head of the construction in which they appear.
6. Adverbs are dependent upon the head of the verb phrase in which they appear.
7. Two-way dependency exists between the head of a phrase and any form of the verb "to be" or the preposition "of." This rule holds for the heads of both phrases linked to these forms.
8. Two-way dependency within or across sentences also exists between tokens of the same noun and between a pronoun and its referent.
9. Dependencies within a passive sentence are treated as if the sentence were an active construction.
10. The head of the subject is dependent upon itself or upon a like token in a preceding sentence.

4.0 Answering Questions

So far we have discussed the kinds of information contained in a question that can be used to identify an answering statement. The content words, the question words and the function words of the question have all been seen to offer important cues which can be used to help identify an answer. In the following two sections we shall describe in detail first, how the information-rich statements which may contain answers are found from searching a large corpus of text, and second, the techniques for actually using the dependency analysis to select from potential answering statements those which are most probably the answers.

4.1 Selecting Information-Rich Text in Response to a Question

A first requirement on any question-answering system is an organized storage of data or the means for obtaining such. In an earlier paper on

maximum-depth indexing (10), Simmons and McConlogue described the Indexer, a system for impressing on running text an organization suitable for extracting potential answers to that text. The Indexer constructs a complete index of all the content words in the text and cross-references such words as "elect," "elections," "electing," "elects," etc. For the Protosynthex I system, a complete index of the Golden Book Encyclopedia was made and this text serves as the basis for the question-answering system.

Questions are input to this system by punched cards although there is a provision for using teletype or flexowriter input if desired. The question may be any combination of English words not to exceed 20 or 25 (depending on the length of each word), followed by a question marker. The first analysis of the question is into the classes, content word versus function or question word. The content words that are extracted are used to find in the index every occurrence in text of a sentence using that word.

When two words such as "farmer" and "election" are looked up in the index, a list of references for each is found. The references are in the form of VAPS numbers (Volume, Article or chapter, Paragraph, Sentence). For "farmer," VAPS numbers for all occurrences of words such as "farm," "farmer," "farms," "farming" are listed under the one form "farmer." A root form logic used in looking up words in the index insures that all alternate forms of the word will lead to that entry. Similarly all the VAPS numbers for "election" and its various forms are recovered from the index.

Thus for a hypothetical (and very vague) question such as "What farmers were elected?" the content words "farmers" and "elected" would have been

selected and the VAPS numbers recovered would represent all sentences in the text that used either or both of the content words in the question. Such a list is illustrated in Table 1. A first step in discovering potential answers is to search for identical VAPS numbers in the lists associated with each of the words. If two words occur in the same sentence, that sentence will be listed as a VAPS number for each of the words. In Table 1, the VAPS number, 1-17-3-1 is common to both words. If no such sentence had been found we would have been interested in any paragraph in which the two words were found. The first VAPS number for each of the words happens to be such a paragraph, i.e., 1-1-5-4, 1-1-5-2.

Entry:	Volume	Article	Paragraph	Sentence
Farmer:	1	1	5	4
	1	1	5	7
	1	17	3	1
	1	17	3	2
Election:	1	1	5	2
	1	17	3	1
	1	35	4	1
	1	35	4	2
	1	36	2	2

Table 1. VAPS Recovery for "Farmer" and "Elected"

By intersecting the sets of VAPS numbers recovered for each of the content words in the question, it is possible to discover those units of text in which some or all of the content words are present. Other things being equal, those sentences, paragraphs, or chapters which contain intersections of the content words are the most likely candidates for answers.

However, other things are not equal in several ways. First, in the case of a one-content-word question such as "What is a farmer?" no intersection is possible. For this case, the paragraph or article containing the most references to that content word is selected. A second case arises from the fact that not all words are equally important for finding an answer.

The distinction between content words and function words is based on the fact that in English some words are primarily used as structural indicators while others carry the semantic content. But there are many differences among content words in the amount of semantic information that they carry. For two extremes, consider the words "type" or "kind" and the word "zebra." The first words are hardly more than markers while the second points precisely to a single meaning. If the question is asked, "What kind of farmer was elected?" the word "kind" changes the meaning to a relatively small extent from the original question. It is hardly to be expected that asking for an intersection of all three content words of the question will bring back any more satisfactory answer than will the two previously considered.

This distinction in importance of content words turns out to be represented at least roughly by their frequency of occurrence in a large sample of text. The most frequently occurring words are of course the functional particles

such as "of," "and," "the," etc. But the most frequent content words include such words as "thing," "type," "kind," etc. In general, the more frequent a content word the vaguer its meaning; the more infrequent the word the more precise its meaning. This is the relationship that is expected from information theory which shows that the symbol that is least probable is generally the one that carries the most information.

The inverse of a word's frequency in a large corpus of text is thus approximately proportional to its importance as an indexing term for finding information-rich statements. Consequently, as a word is looked up in the index, its number of occurrences in the large sample of text is used as its relative frequency count. The inverse of this count for each word serves as a basis for deriving an information score for the question and possible answering statements. For example:

"What kind of farmer was elected?"

Frequency:		30	10	5
Inverse:	1/30 +	1/10 +	1/5 =	1/3 = .333

The sum of the inverse frequencies for the question may be called Qmax. A similar procedure for each answer results in a sum which can be called Amax. The ratio Amax/Qmax is a measure of how closely the information content of words in the answer matches those of the question. Although the statistic is admittedly a first approximation, it has the desired property of weighting most heavily those content words which carry the most information. (In the above example, Qmax would be .300 without the word "kind," and thus an answering statement that contains "farmer" and "elected" would give an information-rich ratio of .300/.333 = .90.

A third case in which the simple intersection is not sufficient occurs when some or all of the content words in the question have no correspondents in the index. For this case a dictionary of synonyms is gradually being developed. Entries for the synonym list are developed empirically by discovering what words in failed questions would have brought back appropriate text. When synonyms are used as part of the lookup query for a question, the scoring for the occurrence of the synonym is attenuated by a large factor which is also obtained by experience with the system.

After the intersecting and scoring phase has been completed, those VAPS numbers which have survived are used to find the actual sentences, paragraphs, or articles which have been estimated to be relevant to the question. As was described in the article on the Indexer (10), the encyclopedia which serves as a text base has been organized on magnetic tape in terms of volumes, articles, and paragraphs. The VAPS numbers serve as addresses to the location of pertinent text and it is a simple matter to spin the tape reel once to retrieve in order all the pertinent statements. The retrieved text along with the questions are then input to the grammar machine for further analysis.

The grammatical system develops a phrase structure analysis for the questions and for each of the proposed answers. A program for transforming from phrase structure analysis to dependency analysis is then run; its output, in terms of what words are dependent on what, becomes the input for the answer evaluation system. Detailed descriptions of these systems have been presented elsewhere by Klein (5). It is here sufficient to comment that in their present versions a good deal of hand editing is required before the resultant dependency is unambiguous enough to use as input for the answer evaluation system.

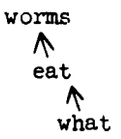
4.2 Recognizing Answers

When we try to discover just what elements of a question are relatively invariant in all answering constructions, it is necessary to make a dependency analysis of both the question and the proposed answering statements. Figure 3 shows such analyses for the examples cited as potential answers to "What do worms eat?" In analyzing this question it is apparent that "worms" is the main noun phrase or subject, "eat" the verb modifying "worms," and "what" is the object modifying "eat." The word "do" belongs to a category of words that are used to signify the question transform and drops from consideration. In Figure 3a the tree structure of this analysis is shown.

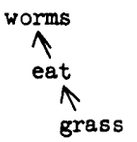
Each of the statements is similarly analyzed for dependencies. Wherever a statement is found in the passive construction, e.g., "Worms are eaten by birds," it is transformed to an active construction--"Birds eat worms." Figure 3b shows the analysis of "Worms eat grass." "Eat" is dependent on "worms," and "grass" is dependent on "eat." The dependency relations of the answer are precisely those of the question with "grass" filling the position of "what." The statement shown in Figure 3c, "Grass is eaten by worms" is the passive equivalent of "worms eat grass." After being transformed to an active construction, its dependencies are identical with those of the question.

In contrast, "Birds eat worms" is shown in 3d. Here "eat" is dependent on "birds," and "worms" is dependent on "eat." Not one of the dependencies in this proposed answer matches those of the question, and the answer may be rejected forthwith.

a) What do worms eat

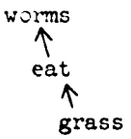


b) Worms eat grass



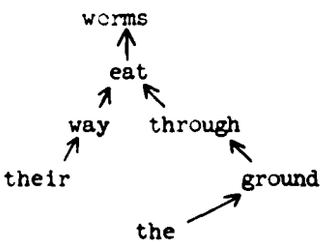
c) Grass is eaten by worms

→Worms eat grass

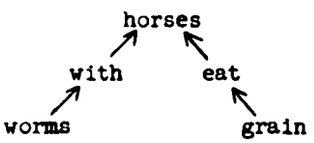


d) Birds eat worms

e) Worms eat their way through the ground



f) Horses with worms eat grain



Dependency Structures of a Question and Some Potential Answers

Figure 3

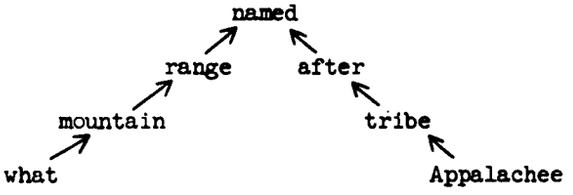
More difficult situations may be examined in 3e and 3f. For the proposed answer "Worms eat their way through the ground," we discover first that "eat" is dependent on "worms" as the question demands. Now do there exist words corresponding to "what" that are dependent on "eat" as in the question? "Way" and "ground" are both nominals and belong to the semantic class "thing" which "what" demands. For "way" the dependency match is again perfect and the sentence is accepted as one that tells us something about what worms eat. In Figure 3f, "Horses with worms eat grain" is readily seen to fail the requirement, "eat" dependent on "worms" and may be rejected. It can be noticed that two senses of "with" are possible in the sentence: "Horses and worms eat grain," or "Horses containing worms eat grain." At present the question evaluation logic is not sensitive to the difference. It might also be argued that if horses with worms inside them eat grain, the worms also eat grain. This inference is also more subtle than can be handled by the question evaluation logic.

These examples show the basic principle of matching dependencies to determine whether or not a statement is in an appropriate structure to form an answer. However, we are not usually so fortunate as these simple examples would indicate. First, identical words are not always to be found or required in potential answering statements. Second, the dependency relationships often are not and need not be perfect matches.

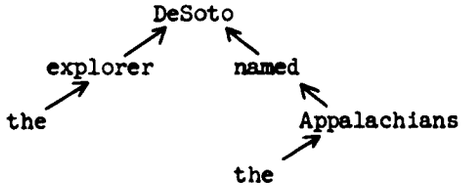
A very difficult example is offered by the question and some proposed answers shown in Figure 4. For the question, "What mountain range was named after the Appalachee tribe?" "what" is required to modify a mountain range and the answer is expected to predicate that this mountain range was named

after the Appalachee tribe. The potential answer is in two sentences. "The explorer DeSoto named the Appalachians. He named them after the Appalachee Indians."

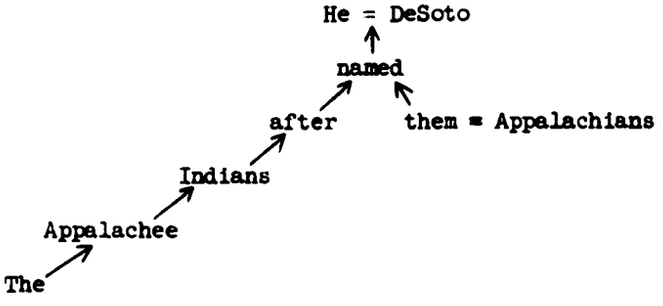
a) What mountain range was named after the Appalachee tribe



b) The explorer DeSoto named the Appalachians



c) He named them after the Appalachee Indians



Dependency Analysis of Complex Structures

Figure 4

The dependency structures of the question and of its proposed answer may best be compared in a matrix which is shown in Figure 5. A "1" in a square of the matrix shows that the word on the left of the row was dependent in the question on the word on top of the column. An asterisk in association with the "1" shows that the proposed answer had the same dependency relation as in the question. It will be noticed first that the question was transformed from passive to active so that it has no subject. It is as though the question had stated "(They) named what mountain range after the Appalachee tribe?" As a consequence, the phrases, "what mountain range" and "after the Appalachee tribe," are both dependent on the verb "named."

The next thing to be noticed in the matrix is that certain words are dependent on others even though they are not adjacent in the dependency tree. This is a consequence of the fact that under many circumstances dependency is transitive. By transitive we mean that if word "A" is dependent on "B" and "B" is dependent on "C," then "A" is dependent on "C." In the present example this means that where "mountain" is dependent on "range" and "range" is dependent on "named," "mountain" is dependent on "named." Similarly, "what" is dependent on "mountain," "range," and "named."

Transitive dependency appears to hold within nominal and verbal constructions but not across the head of the construction if it is a verb, a preposition, or a subordinate conjunction.* Studies of transitive dependency in the context of computer generation of coherent discourse are discussed in

*For some purposes we allow the forms of the verb "to be" and the preposition "of" to be transitive.

earlier papers (5, 6). For question answering, the transitivity of dependence is a primary generalizing feature which allows the recognition of varied forms of English statements as related to the question.

	named	range	mountain	what	after	tribe	Appalachee	(any word)
named								1*
range	1							
mountain	1	1						
what	1*	1	1					
after	1*							
tribe					1			
Appalachee					1*	1		

Note:

1 = dependency of Question

* = dependency of Answer

Matrix Comparison of Question and Answer Dependencies

Figure 5

Returning to the matrix of Figure 4, the asterisks show that the proposed answer held the following dependencies:

"named" on (any word)
"after" on "named"
(what) on "named"
"Appalachee" on "after"

A perfect match of answer to question would have given all of the 11 dependencies recorded from the question. Four matches actually occurred. The ratio 4/11 = .36 offers a basis for scoring the sentence. However, the direct scoring value of this statistic for determining correct answers is not expected to be very strong, and a fair amount of effort will be required to work out the complex probabilities involved in the combined match of words and relations. Such probabilities could be plotted against the judgment of people that a given statement is or is not an answer to a question. We have not yet accumulated enough data in using the system to begin to work out such a detailed scoring procedure.

However, we have not yet exhausted the information contained in the question. In the proposed answer we discover that "Appalachee" is dependent on "Indians." In the question, "Appalachee" was dependent on "tribe." The question, "Indian = tribe?" may be generated. Similarly "Appalachians" in the answer corresponding to the "what" position of the question can be used to generate "Appalachian = mountain range?" These questions can be processed against the synonym dictionary or against the original text. "Indian" can be looked up in the synonym list to discover if one of its alternates is "tribe." Failing this, the text can be searched to discover if there is a statement in which "Indian" is dependent on "tribe." For the second example the same process can be followed to discover if "Appalachian" is dependent on "mountain" and on "range." If these relations are discovered to hold in the larger text, the percentage match is increased, frequently to a rather large extent. The probability that the original statement was an answer would also increase significantly.

In fact the generation of additional questions is not so frequent as the example suggests. When information-rich statements are obtained as potential answers, we frequently find several related statements such as "The Appalachian mountains are located..." or "Tribes of Apache Indians roamed..." Dependencies from these statements, if they are retrieved, are used to complete the matrix of Figure 4 and to avoid the necessity of generating additional questions.

A system which would continually generate smaller and smaller questions in its attempt to answer the original query is conceptually very attractive. One can easily visualize such a system being given a large batch of questions--say twenty--to solve. Each of these questions might result immediately in partial answers plus an average of three to five additional questions. Answering these derived questions might generate some lesser number of questions until finally the system had a certain set for which it could find no answers. At this point the computer could output a request for printed material pertaining to its unanswered questions and so add to its own conceptual capability by increasing its information store. In this fashion, questions could be used to guide the education of a language processor.

From a practical point of view, however, such a system might prove extremely expensive. Each question might generally require several iterations through a large store of information. It is even conceivable that there might be questions which could lead to endless recursions.

A far easier alternate is the system could generate such derived questions for the human operator to answer. His answers would enrich the computer storage in the same manner that reading new text would. Thus, by experience, the computer would continually improve its ability to recognize relationships between words.

It is the latter alternative which was chosen for Protosynthex I. The system examines the matrix which compares question-and-answering text as in Figure 5. It first discovers the fragment of text which is equivalent in dependency structure to the question word "what." This fragment, in the present case, is the single word "Appalachians." It then considers the dependencies of the question word. The word "what" in the question was dependent on "named," "range," and "mountain." The substitutable fragment, "Appalachians" is only dependent on "named." The question, "Is Appalachians dependent on mountain and range?" is output to the human operator. Similarly, the word "Appalachee" is common to both question and answering statement, but in the question "Appalachee" is dependent on both "after" and "tribe," while in the answer "Appalachee" is dependent only on "after." The question generated on the basis of the difference is, "Is Appalachee dependent on tribe?"

A slightly differing form of output has also been contemplated. Given the capability of generating the subsidiary questions above, it is easily possible to print out the following statement:

"If Appalachians are dependent on mountain and range, and Appalachee is dependent on tribe, then the answer is Appalachians."

The underlining refers to the words which the computer has found to be critical in answering the question. This kind of output has the attractive (though trivial) feature of insuring that the computer system will always give a logically correct answer.

5.0 Discussion

It was mentioned earlier that this question-answering logic is part of the prototype language processor, Protosynthex I. The flow of operations in this machine includes indexing text, finding information-rich statements in response to a question, syntactically analyzing both question and potential answers, conversion of this analysis to a dependency analysis and finally evaluation through dependency logic of the set of potential answers. There is one striking weakness in the machine: No existing grammatical analysis system can make a completely automatic unambiguous parsing of text.

Most syntactic analysis systems have been developed in the context of mechanical translation problems. One of the most sophisticated of these is the predictive analysis system of Oettinger and Kuno (8). This system uses a dictionary with a fairly fine classification of English words into parts of speech. Using only syntactic word classes, it discovers dozens of possible interpretations for many if not most English sentences. The protosynthex grammar machine is also plagued by ambiguous interpretations. In our case, a decision is made to select one interpretation to avoid the rapid multiplication of work entailed by the many tree structures possible. Unfortunately there is no way to be certain that the interpretation chosen is the one that best fits the sentence under consideration.

If an analysis that does not fit happens to be selected, the dependency structure that is derived will usually be wrong and the question-answering logic will fail. At this point a human editor must enter the system to correct any errors in the dependency analysis.

At the present state of development of automated grammar machines, the ambiguous interpretations cannot yet be avoided. When a human parses a sentence he does so with full knowledge of the meaning of the words that make it up. The only knowledge the machine can bring to bear is in terms of its word classes and its rules for their combination into phrases, clauses and sentences. The human's knowledge of "meaning" may be considered to be a very much finer set of word classes and combination rules than any computer system has yet been given.

To take a frequently used example, "They are flying planes," let us consider in one case that the referent for "they" is "the Smiths." In another case the referent might be "Cessnas." When a human is given enough context to be able to find the referent, there is no ambiguity. But for existing computer systems either "Cessnas" or "the Smiths" may be flying planes or flying planes (italics to indicate spoken emphasis). For the machines to avoid ambiguity, "Cessna" must belong to a hardware class that includes airplanes and not to the "person" class that includes "the Smiths." In addition, the machine needs grammatical and semantic rules of combination such that the hardware class can be combined with objects that act but not with a certain set of actions such as instigating the act of flying.

In actual fact no such detailed analysis has ever been made of English or any other natural language to date.* Perhaps it never will be made if computers that learn to process English as humans do can be developed. But

*However, S. Lamb at University of California, Berkeley in his sememic analysis approach to mechanical translation is working toward such an analysis.

without such a fine-grained semantic classification, whether it be formally inserted into a program's tables or be developed by a learning system, the problem of ambiguity will remain a central one in all efforts at fairly sophisticated language processing.

For any small set of language--a few hundred to a few thousand words--it may be possible that a grammatical and semantic system can be developed which will reduce the ambiguity of analysis to manageable proportions. Such a system would undoubtedly include hundreds of word classes and numerous special rules of combination. It would be especially designed to account for the particular small selection of English on which it was developed but it would generalize to other texts providing they used only the same vocabulary and similar constructions.

Building such a small-scale system would teach us much about methods for constructing semantically sophisticated machines which are already needed for the various types of computer language processing. In Protosynthex I, a hand-editing stage is required to correct the errors in dependency analysis caused by ambiguous grammatical interpretations. This stage could be eliminated in the small-scale language processor and a completely automatic, high-quality system for answering questions on a limited sample of text would come into existence.

References:

1. Chomsky, N. Syntactic Structures. 's-Gravenhage: Mouton and Co., 1957.

2. Green, B. F. Jr., Wolf, A. K., Chomsky, C., and Laughery, K. Baseball: An Automatic Question Answerer. Proceedings of the Western Joint Computer Conference, Vol. 19, pp. 219-224, 1961.

3. Harris, Z. S. Co-occurrence and Transformation in Linguistic Structure. Language, Vol. 33, No. 3, pp. 283-340, 1957.

4. Hays, D. G. Studies in Machine Translation--10: Russian Sentence-Structure Determination, RM-2538. The RAND Corporation, 1960.

5. Klein, S. Automatic Decoding of Written English. Unpublished doctoral dissertation, University of California, Berkeley, 1963.

6. Klein, S. and Simmons, R. F. Syntactic Dependence and the Computer Generation of Coherent Discourse. Mechanical Translation, in press. (Also available as SDC document TM-758/000/00.)

7. Lindsay, R. K. The Reading Machine Problem. Unpublished doctoral dissertation, Carnegie Institute of Technology, 1960.

8. Oettinger, A. G. Mathematical Linguistics and Automatic Translation. Report to the National Science Foundation, No. NSF-8, Cambridge, Massachusetts, January 1963.

9. Rankin, B. K. A Programmable Grammar for a Fragment of English for Use in an Information Retrieval System. National Bureau of Standards Report No. 7352, June 1961.

10. Simmons, R. F. and McConlogue, K. Maximum-depth Indexing for Computer Retrieval of English Language Data. American Documentation, Vol. 14, No. 1, pp. 68-73, 1963. (Also available as SDC document SP-775.)

11. Yngve, V. H. A Model and an Hypothesis for Language Structure. Proceedings of the American Philosophical Society, Vol. 104, No. 5, pp. 444-466, 1960.

UNCLASSIFIED

System Development Corporation,
Santa Monica, California
CO-OCCURRENCE AND DEPENDENCY LOGIC
FOR ANSWERING ENGLISH QUESTIONS.
Scientific rept., SP-1155, by
R. F. Simmons, S. Klein, K. McConlogue.
3 April 1963, 30p., 11 refs., 5 figs.
(Contract SD-97)

Unclassified report

DESCRIPTORS: Machine Translation.
Language.

States that one of the most attractive approaches to language processing UNCLASSIFIED

---

on computers is the research on program systems that enable the computer to answer natural English questions. Discusses the kinds of information contained in a question that can be used to identify an answering statement. The content words, the question words and the function words of the question have all been seen to offer important cues which can be used to help identify an answer. Describes how the information-rich statements which may contain answers are found from searching a large corpus of text, and the techniques for actually using the dependency analysis to select from potential answering statements those which are most probably the answers. UNCLASSIFIED